OCR Computer Science A Level

2.1.1 Thinking Abstractly

Advanced Notes

Specification:

2.1.1 a)

- The nature of abstraction

2.1.1 b)

- The need for abstraction

2.1.1 c)

- The difference between abstraction and reality

2.1.1 d)

- Devise an abstract model for a variety of situations

The nature of abstraction

Abstraction is one of the most important principles in Computer Science and is a critical part of computational thinking. It is the process of removing excessive details to arrive at a representation of a problem that consists of only the key features. Abstraction often involves analysing what is relevant to a given scenario and simplifying a problem based on this information. This is called representational abstraction.

Another form of abstraction involves grouping together similarities within a problem to identify what kind of problem it is. This is called abstraction by generalisation and allows certain problems to be categorised as being of a particular type. Thus a common solution can be used to solve these problems.

Data abstraction is a subcategory of abstraction in which details about how data is being stored are hidden. As a result, programmers can make use of abstract data structures such as stacks and queues without concerning themselves with how these structures are implemented.

Synoptic Link

You will have come across **abstract data structures**, their characteristics and functionality in **1.4.2.**

Programmers can also perform functions such as pushing and popping items to and from a stack without having any knowledge about the code used to implement this functionality. This is called procedural abstraction and is also used in decomposition. It models what a subroutine does without considering how this is done. Once a procedure has been coded, it can be reused as a black-box.

Very large, complex problems make use of multiple levels of abstraction, where each level performs a different role. The highest levels of abstraction are closest to the user and are usually responsible for providing an interface for the user to interact with hardware whereas the lowest levels of abstraction are responsible for actually performing these tasks through the execution of machine code.

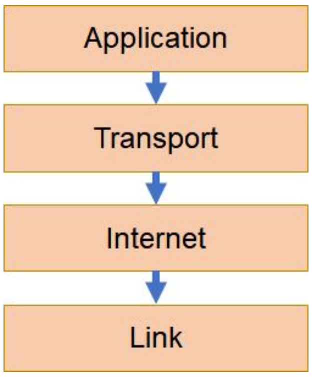
The need for abstraction

At its core, abstraction allows non-experts to make use of a range of systems or models by hiding information that is too complex or irrelevant to the system's purpose.

Abstraction enables for more efficient design during software development as programmers can focus on elements that need to be built into the software rather than worrying about unnecessary details. This then reduces the time needed to be spent on the project. Removing wasteful details early on also prevents the program from getting unnecessarily large.

Layers of abstraction are used within networking and programming languages. Programming languages can be separated out into a spectrum of high and low-level languages. Low-level languages such as assembly code and machine code directly interact with computer systems but are more difficult to write. Programming using machine code requires having an understanding of the functions specific binary codes perform and although assembly code is easier to memorise, it still requires programmers to know the mnemonics associated with the instruction set specific to the processor. High-level languages provide an abstraction for the machine code that is in fact executed when a program is run. This makes the process of developing programs easier, as syntax in high-level languages parallels natural language and is considerably easier to learn and use compared to low-level languages. This has also made coding accessible to non-specialists.

The TCP/IP model is an abstraction for how networks function, separated into four layers of abstraction: application, transport, internet and link. Each layer deals with a different part of the communication process, and separating these stages out makes them simpler to understand. Each layer does not need to know how other layers work. Outgoing communication is visualised as going down these layers, while incoming information can be imagined as going up these layers. However, it is also important to ensure compatibility between these layers so standards must be agreed in advance. The TCP/IP model uses a set of protocols which means that each layer can be dealt with individually, with details about other layers being hidden.



Synoptic Link

The **TCP/IP model** and the protocols associated with each layer are discussed in detail in **1.3.3**.

The difference between abstraction and reality

Abstraction is a simplified representation of reality. Real-world entities may be represented using computational structures such as tables and databases. Real-world values are often stored as variables.

Synoptic Link

1.2.4 explains the techniques used in **OOP** in more detail.

Object-oriented programming makes use of objects, which are also an abstraction for real-world entities. In object-oriented programming, abstraction considers the functionality, interface and properties of entities. Attributes are an abstraction for the characteristics of an object while methods are an abstraction for the actions a real-world object is able to perform.

Devise an abstract model for a variety of situations

When devising an abstract model given a scenario, you must consider:

- What is the problem that needs to be solved by the model?

  Can the problem be solved computationally? What are the key features of the problem?

- How will the model be used?

  What sort of format does the model need to be displayed in? Consider factors such as convenience, affordability and ease of access.

- Who will the model be used by?

  How many people will be using the model? What level of expertise do they have in the subject/ discipline associated with the problem?

- Which parts of the problem are relevant based on the target audience and the purpose of the model?

  Remove sections that are not relevant to the problem that needs solving. Remove details that will confuse the audience.